

RECEIVED
CENTRAL FAX CENTER

AUG 04 2006

CLAIM AMENDMENTS

This listing of claims replaces all prior versions and listings of the claims in this application:

1. (Currently Amended) A distributed database caching system for processing transactions between an application server and a central DBMS server, the system comprising:
 - a resource abstraction layer on the application server; and
 - a remote server unit in communication with the application server and a database subscription manager, wherein the remote server unit includes a cache DBMS server to manage a cache database, and wherein the database subscription manager is in communication with the central DBMS server,

wherein the application server sends queries for a plurality of users to the remote server unit via the resource abstraction layer,

wherein the remote server unit processes each query through the cache DBMS server, and wherein the cache DBMS server checks a data structure consisting of subscribed query predicates, and wherein, if the query is contained within prior query predicates, the remote server unit sends the local query results result derived from the cache database to the application server, and wherein if the query is not contained within subscribed predicates, the remote server unit sends the query to the data subscriber manager,

wherein the database subscription manager retrieves a result from the central DBMS server, and wherein the database subscription manager derives the query results from the central DBMS server, and wherein the database subscription manager sends the query results to the remote server unit, and creates a subscription to the query predicate on behalf of the remote server unit,

wherein the query results are added to the cache database and the query predicate is added to the query predicate data structure on the remote server unit, completing a subscription to that query predicate.

2. (Original) The system of claim 1, wherein the resource abstraction layer is embedded in a database application programming interface.

3. (Original) The system of claim 2, wherein the database application programming interface is a JDBC client driver.

4. (Original) The system of claim 2, wherein the database application programming interface is a ODBC client driver.

5. (Original) The system of claim 1, wherein the resource abstraction layer comprises a distribution policy, wherein the distribution policy directs the queries to one of a plurality of remote server units.

6. (Original) The system of claim 5, wherein the plurality of remote server units are housed in a data center with the application server and the DBMS server.

7. (Original) The system of claim 5, wherein the plurality of remote server units are housed in a data center apart from the application server and the DBMS server.

8. (Original) The system of claim 5, wherein a first portion of the plurality of remote server units are housed in a data center with the application server and the DBMS server, and wherein a second portion of the plurality of remote server units are housed in a location geographically remote from the data center.

9. (Original) The system of claim 1, wherein the remote server unit further comprises:

a database server driver comprising a database listener function, a query parser function and a database query fulfillment function; and

a query execution manager comprising a query execution function, and wherein the database listener function receives the queries from the application server and passes the queries to the query parser function, and wherein the query parser function parses the queries into structured query language queries and passes the structured query language queries to the query fulfillment function, and wherein the query fulfillment function interacts with the query execution function to obtain the result from the cache DBMS server.

10. (Original) The system of claim 9, wherein the cache DBMS server comprises a cache manager function, a cache description handler function, a cache containment function, a notification processor, and a space manager, wherein the cache manager function interfaces with the query parser function and the cache containment function to determine whether the structured query language queries can be satisfied from the cache DBMS server, and wherein the cache description handler updates the subscribed query predicates and provides a predicate usage information to the space manager, and wherein the notification processor receives notification messages from the database subscription manager and propagates an effect of the notification messages to a cache database on the cache DBMS server.

11. (Original) A distributed database caching system for processing transactions between an application server and at least one central DBMS server, the system comprising:

a modified database client driver on the application server, wherein the modified database client driver includes a resource abstraction layer;

an application logic on the application server, wherein the application logic sends a user inquiry to the modified database client driver for processing and wherein the application logic receives a result from the modified database client driver; and

at least one remote server unit in communication with the application server and at least one database subscription manager, wherein the remote server unit includes a modified database server driver, a cache DBMS server, and a query engine, and wherein the cache DBMS server includes a cache database and a data structure containing currently subscribed query predicates, and wherein the at least one database subscription manager is in communication with the least one central DBMS server,

wherein the resource abstraction layer receives the user inquiry from the modified client database driver and sends the user inquiry to the modified server database driver on the at least one remote server unit,

wherein the modified server database driver sends the user inquiry to the cache DBMS server and wherein the modified server database driver receives the result from the cache DBMS server,

wherein the cache DBMS server uses the query engine to check the predicate data structure for a locally contained result, and wherein, if the local result exists, the cache DBMS server sends the local query result to the modified server database driver as the result, and wherein if the local result is nonexistent, the cache DBMS server sends the user inquiry to the at least one data subscription manager,

wherein the at least one database subscription manager retrieves the result from the at least one central DBMS server, and wherein the at least one database subscription manager derives a plurality of predicates from a plurality of user inquiries processed by the at least one central DBMS server, and wherein the at least one database subscription manager sends the plurality of predicates to the cache DBMS server for updating the cache database.

12. (Original) A method of processing a transaction between an application server and a central DBMS server, the method comprising:

sending the transaction from the application server to a remote server unit, wherein the transaction is sent via a resource abstraction layer on the application server;

determining whether a local result exists in a cache database on the remote server unit;

sending the local result from the remote server unit to the application server if the local result exists;

sending the transaction from the remote server unit to a data subscriber manager if the local result does not exist, wherein the database subscription manager retrieves a result from the central DBMS server;

deriving on the database subscription manager a plurality of predicates from a plurality of transactions processed by the central DBMS server;

sending the plurality of predicates from the database subscription manager to the remote server unit; and

updating the cache database according to the plurality of predicates.

13. (Original) A method of implementing serializable transaction isolation in a distributed database caching system in communication with a central DBMS server, the method comprising:

opening a serializable transaction on an application server;

placing an entry into a local update request queue on a remote server unit;

opening a remote transaction on a central DBMS server using a database subscription manager in response to an update request from the remote server unit;

sending a plurality of updates from the database subscription manager to the remote server unit using data obtained from the central DBMS server;

processing the plurality of updates in a serialized manner on the remote server unit by checking the plurality of updates to identify predicates in the cache database, locking the predicates in the cache database, and starting a local transaction on a remote server unit; thereby

synchronizing the remote transaction and the local transaction.

14. (Original) A method of implementing read committed transaction isolation in a distributed database caching system in communication with a central DBMS server, the method comprising:

opening a read committed transaction on an application server;

starting a local transaction on a remote server unit;

receiving a user request on the remote server unit;

opening a remote transaction on a central DBMS server using a database subscription manager in response to a data request from the remote server unit, wherein the data request is based on the user request;

sending a result from the database subscription manager to the remote server unit using data obtained from the central DBMS server;

sending the result from the remote server unit to the application server;

receiving a commit request on the remote server unit from the application server;

sending the commit request to the database subscription manager; and

verifying a successful outcome for the commit request.

15. (Original) The method of claim 14, wherein the read committed transaction comprises a repeatable read transaction, and wherein the method further comprises sending plurality of timestamp messages from the data management unit to the remote server unit on a periodic basis, wherein the remote server unit synchronizes with the central DBMS using the timestamp of the remote server unit transaction to process data requests on the central DBMS.

16. (Original) A method to determine the impact of changes to data on a central DBMS server on subscribed predicates on individual distributed cache systems, the method comprising:

mapping rows and associated subscriptions with a unique identifier for each row;

mapping column values to subscribed predicates,

checking the unique identifier of a changed row for containment within subscribed

rows,

checking the column values of changed rows for containment in subscribed

predicates,

notifying affected distributed cache systems to implement the changed rows as

updates.

17. (Original) A method of implementing serializable transaction isolation in a distributed database caching system in communication with a central DBMS server, the method comprising:

- opening a pool of serializable transactions on the central DBMS server;
- associating each pooled transaction with updated information on the central DBMS server,
- sending the updated information and an identifier for each pooled transaction to a remote server unit,
- opening a serializable transaction on the remoter server unit and processing the updated data associated with it,
- using the open serializable transactions opened on the remote server unit and the central database server to process read and write requests to the remote server unit,
- processing a final transaction commit on the central DBMS server,
- sending one of a rollback or a successful commit message to the remote server unit.